**Decision Tree step by Step:**

**What is Decision Tree?**

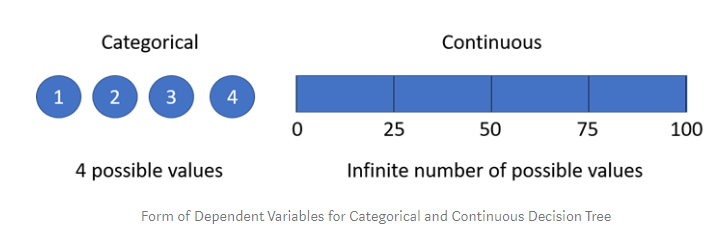
A decision Tree is tree-like structure that is used to model for classifying data. It decomposes in sub trees made up of nodes/leaf

**Why Decision Tree?**

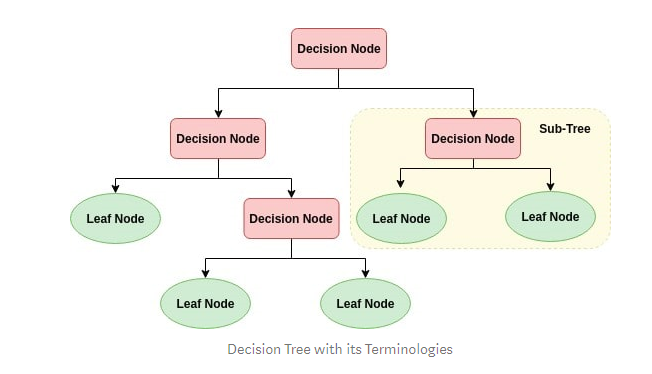
A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

The feature importance is clear and relations can be viewed easily.

It can be utilized for regression and classification, so called as CART-Classification And Regression Tree.



**How Decision Tree look like?**



Root Node (Top Decision Node): It represents the entire population or sample and this further gets divided into two or more homogeneous sets.

Splitting: It is a process of dividing a node into two or more sub-nodes.

Decision Node: When a sub-node splits into further sub-nodes, then it is called a decision node.

Leaf/ Terminal Node: Nodes with no children (no further split) is called Leaf or Terminal node.

Pruning: When we reduce the size of decision trees by removing nodes (opposite of Splitting), the process is called pruning.

Branch / Sub-Tree: A sub section of the decision tree is called branch or sub-tree.

Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

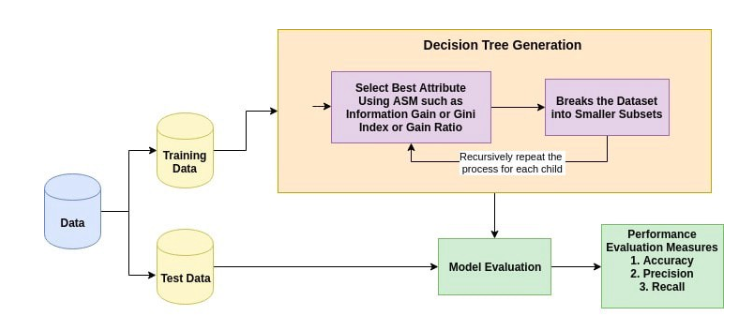
**How does Decision Algorithm Works?**

Select the best attribute using Attribute Selection Measures(ASM) to split the records.

Make that attribute a decision node and breaks the dataset into smaller subsets.

Starts tree building by repeating this process recursively for each child until one of the condition will match:

* All the tuples belong to the same attribute value.
* There are no more remaining attributes.
* There are no more instances.



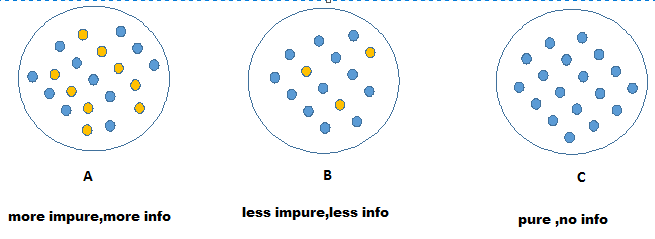
**How Attributes selection Method?**

ASM(Attribute selection Method) gives rank to each feature by explaining given dataset.Best score attribute will be selected as Splitting Attribute. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node.

**1.Entropy**:It means how much variance data has.

Ex.A dataset has only blue balls. Means zero Entropy because it is pure dataset.

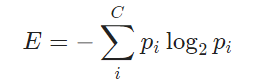
A dataset with mix balls of blue, green, red would have relatively high entropy.



Entropy(S)=0, if pure dataset

Entropy(S)=1 , if mixed samples are of same number

Entropy(S)=0 to 1, if mixed samples are of various numbers



Ex: The easiest way to understand this is, 1 blue,2 greens,3 reds. Then,

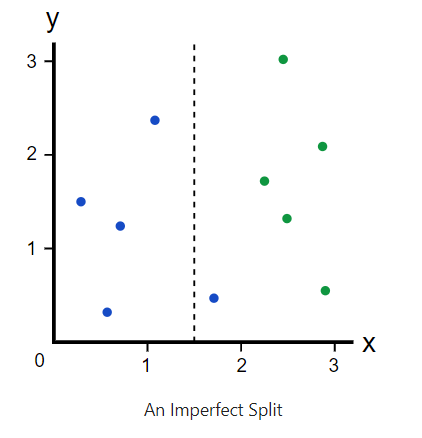
E = −(*pb*​log2​*pb*​+*pg*​log2​*pg*​+*pr*​log2​*pr*​)

​=−((1/6) ​log2​(1/6​)+(2/6)​)log2(2/6​)+(3/6)log2​(3/6​))=1.46​​

What about dataset of one color, means 3 blue balls

*E*=−(1log2​1)=0​

**2.Information Gain**-It is used to measure quality of split

. 

Before split, there are 5 blues and 5 greens, so entropy was

*IG(Ebefore)*​​=−(0.5log2​0.5+0.5log2​0.5)=1​​

After the split, we have two branches. Left branch has 4 blue balls, so

IG(Eleft)=0, because it has all same color.

Right branch has 1 blue ball and 5 greens, so

*IG(Eright)*​​=−(61​log2​(61​)+65​log2​(65​))=0.65​​

Now, we have the IG for both branches, we can determine the quality of split by weighing the entropy of each branch by how many elements it has. Since left branch has 4 elements and right branch has 6 elements , we weigh them by 0.4 and 0.6 resp.

IG(E split)= probability of (Left)\* IG of left +probability of (Right)\* IG of Right

IG(E split)=0.4 \* 0 + 0.6 0 \* 0.65=0.39

Information Gain=How much entropy we removed

Gain=1-0.39=0.61

Higher Information Gain=more Entropy removed

3.Gini-It is used to binary split of tree, mostly success or failure, yes/no

It is sum of squares of probability of success and failure

Gini of split= =square(p)+ square(q)

Gini of impurity = 1 - Gini of split

**Example**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Age** | **Competition** | **Type** | **Profit** |
| Old | yes | software | down |
| Old | no | software | down |
| Old | no | hardware | down |
| mid | yes | software | down |
| mid | yes | hardware | down |
| mid | no | hardware | up |
| mid | no | software | up |
| new | yes | software | Up |
| new | no | hardware | Up |
| new | no | software | Up |

* Step 1:Find out Information Gain for each class of Age. So, new table will become as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Pi | Ni | IG(Pi,Ni) |
| old | 0 | 3 |  |
| mid | 2 | 2 |  |
| new | 3 | 0 |  |

IG(Pi,NI) =-P/P+N log(P/P+N) – N/P+N log(P/P+N)

IG(old) = -(0/10)log(0/10) – (3/10)log(3/10)=0

IG(mid) = -(2/10)log(2/10) – (2/10)log(2/10)=1

IG(new) = -(3/10)log(3/10) – (0/10)log(3/10)=0

Entropy of Age = Pi+Ni/P+N \* IG(Pi,Ni) - (for all classes of Age)

=(0+3/5+5 \*0) + (2+2/5+5 \*1) +(3+0/5+5 \*0)

=0.4

class entropy means (down=5,up=5, so entropy of profit/class=1)

Gain =class entropy – Entropy of Age

= 1- 0.4

=0.6

* Step 2 : Find out Information Gain for each class of Competition. So, new table will become as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Competition | Pi | Ni | IG(Pi,Ni) |
| Yes | 1 | 3 | 0.81127 |
| No | 4 | 2 | 0.918295 |

IG(Pi,Ni) = -P/P+N log(P/P+N) – N/P+N log(P/P+N)

IG(yes) = -1/4log(1/4) -3/4log(3/4) = 0.81127

IG(no) =-4/6 log(4/6) – 2/6log(2/6)=0.918295

Entropy of Competition = Pi+Ni/P+N \* IG(Pi,Ni)

=1+3/5+5 \* (0.81127) + 4+2/5+5 \*(0.918295)

=0.8754

Class entropy =1 (5 up and 5 down in profit)

Gain = class entropy – Entropy of competition

=1-0.8754

=0.124515

* Step 3 : Find out Information Gain for each class of Type.So, new table will become as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Pi | Ni | I(Pi,Ni) |
| software | 3 | 3 | 1 |
| hardware | 2 | 2 | 1 |

IG(Pi,Ni) = -P/P+N log(P/P+N) – N/P+N log(P/P+N)

IG(software) = -3/6log(3/6) -3/6log(3/6) = 1

IG(hardware) =-2/4 log(2/4) – 2/4log(2/4)=1

Entropy of Type = Pi+Ni/P+N \* IG(Pi,Ni)

=3+3/5+5 \* (1) + 2+2/5+5 \*(1)

=1

Class entropy =1 (5 up and 5 down in profit)

Gain = class entropy – Entropy of type

=1-1

=0

Step 4: Compare Gain of Age, Competition, Type to decide which will become root node.

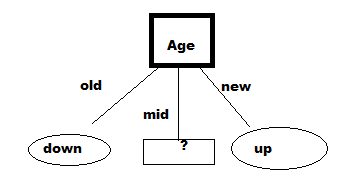
Age has larger value of Gain, so it will become root Node and divides branches into its class labels.

Old has only profit class as down, so down will become leaf node.

New has only profit class as up, so up will become leaf node.

Mid has mixed down and up class labels, so it will further divide into nodes.

|  |  |
| --- | --- |
| Age | 0.6 |
| Competition | 0.12 |
| Type | 0 |



* Step 5: We have to repeat step 1 to 4 for this new table as below, to find out sub nodes of mid branch.

|  |  |  |  |
| --- | --- | --- | --- |
| Age | competition | Type | Profit |
| mid | yes | software | Down |
| mid | yes | hardware | Down |
| mid | no | hardware | Up |
| mid | no | software | Up |

Entropy of class/profit = 1 (because down=2,up=2)

* Step 6: Find out Entropy of competition from above table.

|  |  |  |  |
| --- | --- | --- | --- |
| competition | Pi | Ni | I(Pi,Ni) |
| yes | 0 | 2 | 0 |
| no | 2 | 0 | 0 |

IG(Pi,Ni) = -P/P+N log(P/P+N) – N/P+N log(P/P+N)

IG(yes) = -0/2log(0/2) -2/2log(2/2) = 0

IG(no) =-2/2 log(2/2) – 0/2log(0/2)=0

Entropy of competition = Pi+Ni/P+N \* IG(Pi,Ni)

=0+2/2+2\* (0) + 2+0/2+2\*(0)

=0

Class entropy =1 (5 up and 5 down in profit)

Gain = class entropy – Entropy of competition

=1-0

=1

* Step 7: Find out Entropy of Type from above table

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Pi | Ni | I(Pi,Ni) |
| software | 1 | 1 | 1 |
| hardware | 1 | 1 | 1 |

IG(Pi,Ni) = -P/P+N log(P/P+N) – N/P+N log(P/P+N)

IG(yes) = -1/2log(1/2) -1/2log(1/2) = 1

IG(no) =-1/2 log(1/2) – 1/2log(1/2)=1

Entropy of type = Pi+Ni/P+N \* IG(Pi,Ni)

=1+1/2+2\* (1) + 1+1/2+2\*(1)

=1

Class entropy =1 (5 up and 5 down in profit)

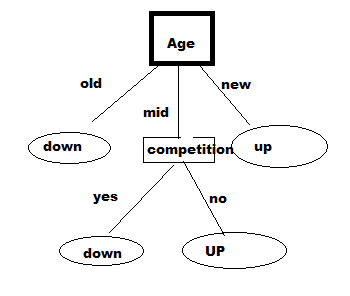
Gain = class entropy – Entropy of type

=1-1

=0

Step 8: Compare gain and decide next node.As competition has higher value, it will become node.

|  |  |
| --- | --- |
| Competition | 1 |
| Type | 0 |



Important Links : <https://towardsdatascience.com/machine-learning-basics-descision-tree-from-scratch-part-i-4251bfa1b45c>